Decadal Survey and Reuse

- If we were to do EOS over again, what would we have done differently?
  - Implement a mission and algorithm software (and documentation) repository
    - Current repository is distributed
    - This could be solved by developing a set of best practices (and then evolve these practices into standards and policies) – is there a best practices document for software reuse? – See "reuse guideline" documents on bottom up reuse (on Reuse portal).
    - Science community needs to on the same wavelength – needs to involve data systems folks in the discussion of future planning for data system policies
  - Don't be so ambitious – too many promises about great leaps forward
  - Spend more on data systems (of overall mission budget) – algorithm development will cost more and take longer than expected
  - Overall structure should be less monolithic – it started that way but became more distributed (i.e. SIPS) – but this may have had a detrimental impact on preserving algorithms and software – the requirements for the SIPS should include requirements for preserving (and delivering) algorithms and software
  - Lesson learned: it will be a few years after launch before good quality data is produced and that revisions will be needed for the life of the mission
  - Algorithm documentation need to be updated before/after major changes to the software
  - Standards and protocols are needed for preserving algorithms and software in a distributed system
  - What are the high leverage points (in terms cost and schedule) for software reuse in the decadal survey missions? Studies are needed to figure this out.
  - Future reuse (persistent use) is important (mother and apple pie)

(cont’d)

- What role should Service Oriented Architecture (SOA) and other new technologies play?
  - Cloud computing should be considered as an option – but data ownership and control is an key issue – there are large potential cost savings – but we need aware other risks – through service level agreements (SLAs) – this could be done within (internal to) NASA
  - SOA – if done right, it will have a large cost savings in the future – but raises new issues (risks) – there may be provenance issues – trust but verify (often) – science testing in a known environment (to verify SOA results) is needed to produce good quality products
  - Verification of correct results from SOA is different and more difficult – many more independent changes are possible – different approach to verification may be needed (self checking)
  - Reproducibility could be a problem – how do you later reproduce a production run after all of the versions of the intermediate steps have changed? This could be handled by “snap shots” of end-to-end change at specific points in the process.
  - Preserving the software (for persistent use or future reuse) is more difficult
- What should be done to prepare for the decadal survey missions?
  - Input data from outside organizations being used in the algorithm needs to be perserved (either by outside organization or internally) – this is also an issue for current systems – reproducibility issue
  - RES(s) – either central or distributed should be established – but links are needed between distributed RESs
  - Linkages between data and software to produce data (provenance) should be established and maintained
  - Open Source software should be used whenever feasible – because of the risks of proprietary software – this helps with preservation of software
  - Policies for use of web services may need to be established – wrt. software and algorithm capture and preservation
  - New process for prioritization of algorihtms and data sets is needed (community vetted)